

## Appendix A

### The Lightning Data Transport I/O Bus Architecture

TOP SECRET



API NetWorks, Inc.  
LDT I/O White Paper  
Revision # 1001

## The Lightning Data Transport I/O Bus Architecture

12.8 Gigabyte/second Bandwidth I/O Bus Solution Builds on PCI Bus  
Concepts to Deliver High Performance for Embedded Systems

### Executive Summary

High performance networking and communications systems, digital consumer electronics, information appliances, set-top boxes, and office automation applications are employing high speed, low power 32- and 64-bit embedded processors to provide specialized functionality at a reduced system cost. With embedded processor technology reaching 1 GHz and up clock frequencies, embedded system developers are facing a new problem: speedy processors need equally fast I/O and multiprocessor busses to keep pace.

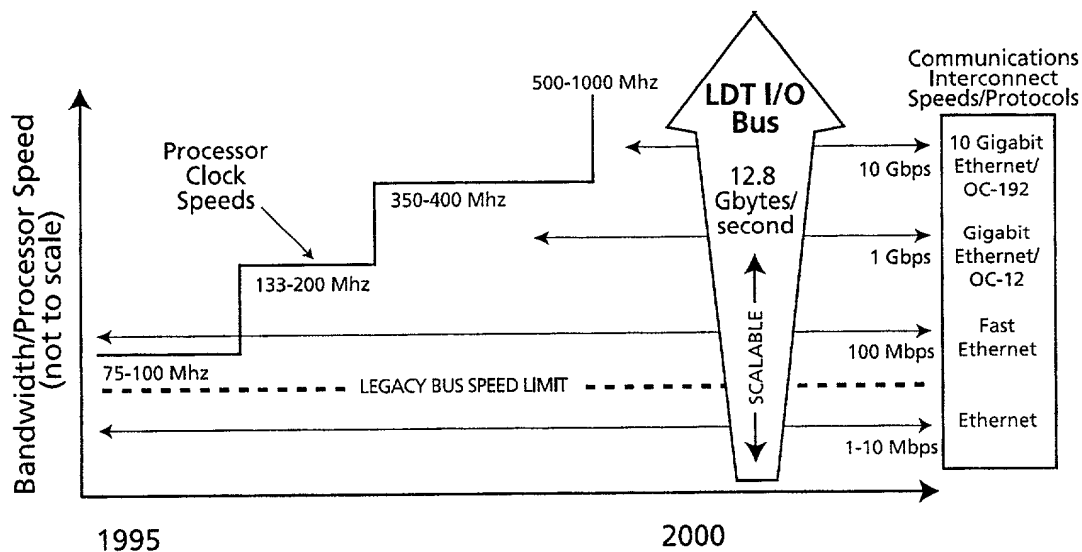
Compact processors and tightly coupled memory busses are providing compute engines with far greater horsepower than ever before. This provides the compute power to perform high speed processing of complex multimedia data, to handle communications and compression algorithms and to manage complex routing databases and addressing tasks. In addition, communications protocols for connecting systems and networks are advancing the pace of data transfers from the Ethernet standards of the past to the 10 Gigabit and OC-192 speeds of the near future.

Legacy I/O bus architectures are widely used in embedded systems because they are low cost and easily implemented using established software and hardware standards. But, these busses top out at 66 MHz or so. Today's processors operating at 500 MHz and 1 GHz and up clock frequencies need a faster alternative to these low bandwidth busses.

The LDT I/O bus delivers the high bandwidth needed for high performance applications in networking, communications and other embedded applications in a flexible, extensible and easily implemented bus structure. A scalable solution, the LDT I/O bus is capable of providing the bandwidth for next generation processors and communications systems. It is a multivendor standard that is easily implemented. The LDT solution provides a broad selection of bus widths and speeds that can fit the power, space and cost requirements of

a wide range of embedded systems from low cost desktop workstations to digital consumer applications and communication systems and networking equipment.

With the ability to scale from narrow configurations with relatively low speed (200 MHz) clock rates to upwards of 32-bit wide, high speed (800 MHz and up) clock rates, the LDT I/O bus architecture is the ideal platform for implementing the next generation of embedded systems.



**Figure 1 – The Lightning Data Transport I/O Bus or LDT provides the bandwidth needed for today's high speed processors and communication systems by delivering up to 12.8 Gigabyte/second bandwidth in a wide variety of scalable I/O bus implementations.**

## LDT Overview

API NetWorks, Inc. and its technology partner, AMD, developed the Lightning Data Transport (LDT) I/O bus structure to solve the I/O bottleneck in high performance 32- and 64-bit processor based systems. LDT delivers a scalable architecture that provides better than an order of magnitude increase in bus transaction throughput over existing I/O bus architectures such as PCI, PCI-X and AGP and compares favorably with newly proposed I/O bus structures such as RapidIO™ and Infiniband. From the system design point of view, the LDT I/O bus provides for the same type of tree of buses as the widely used PCI bus, and uses the same ordering rules as PCI. Best of all, it provides high performance throughput while maintaining backward compatibility with existing software developed for the PCI bus, including the ability to support memory read/write operations.

PCI software driver compatibility is a critical factor for the myriad of developers with investments in PCI-compatible driver software. During the past decade, the PCI bus has been called the universal socket because of its widespread use in everything from

expansion buses for single-chip controllers to use as the standard I/O bus in desktop PCs and as a mezzanine bus in a number of industrial and embedded systems. The fact that software bus drivers for LDT-based I/O systems can use PCI driver software will lower the cost of entry for a wide variety of providers of I/O and subsystem Intellectual Property or IP. This in turn will make available a wider universe of functions for LDT-based system developers, as compared to those using other proposed high-speed buses.

The flexibility of the LDT I/O bus architecture makes it a comprehensive solution to the needs of today's embedded systems. Clock rates range from 200 MHz to 800 MHz. Standard bus widths of 2, 4, 8, 16, and 32 bits can be employed to tailor the I/O bus characteristics to a specific application. Inherent LDT flexibility includes asymmetric bus widths to support different upstream and downstream bandwidth requirements. Along with each data link is a corresponding control line and one or more clock signals. With a fully implemented dual 32-bit wide data link, the LDT I/O bus can deliver 6.4 Gigabytes of bandwidth in each direction. Even with a simple, dual two-bit wide data link, LDT delivers 400 Mbytes/second in each direction.

This bandwidth is exactly what the next generation of embedded systems will require. For example, a 16-bit LDT I/O bus delivers 25.6 Gbits/second bandwidth capable of supporting two OC192 SONET bit streams, or two 10Gbit/second Ethernet links.

With the ability to support point to point links and network links, using LDT switching fabrics, the LDT I/O bus architecture provides a universal building block structure for implementing complex, high speed systems and simple, single function I/O blocks.

In addition to flexibility, scalability and very high bandwidth, LDT also offers the potential benefit of a very low implementation cost. Because it is part of the AMD processor roadmap for its high volume, desktop PCs and server systems, it will benefit from the economies of scale that come with being a part of the high volume, cost sensitive PC-driven market space. This will be similar to the phenomena experienced as PCI adoption grew beyond the PC market.

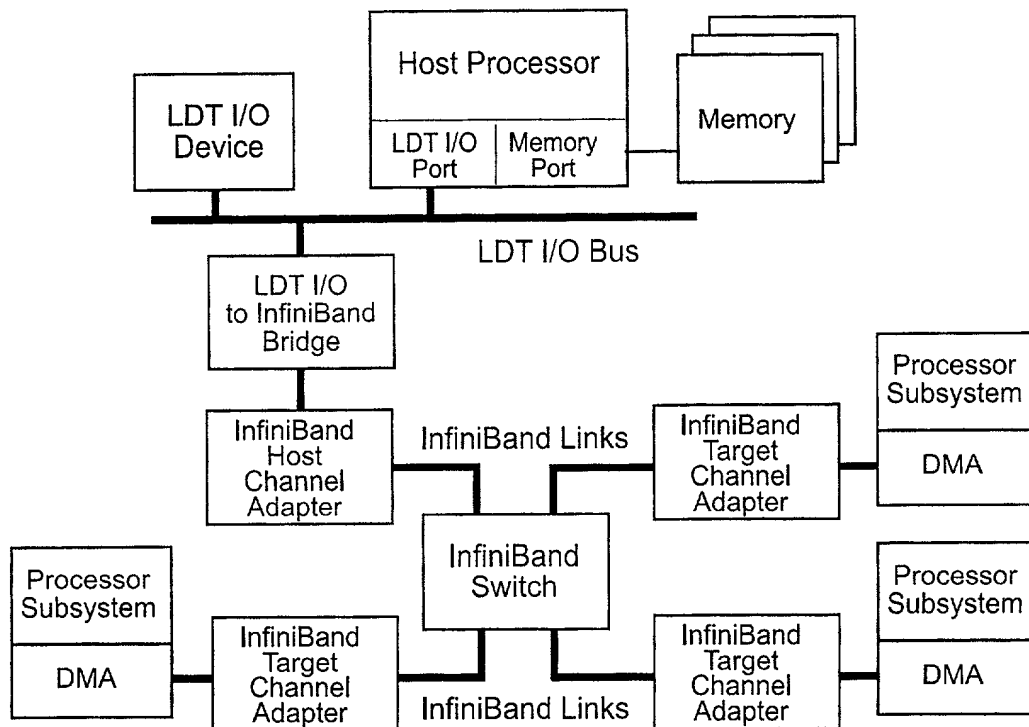
This combination of performance, extensibility and low cost of implementation makes the LDT bus an attractive real-world option for the developer of high performance embedded systems, including those used in network routing and switching, packet processing, smart communications equipment, high end digital entertainment and information appliances.

### **LDT I/O Bus, RapidIO and InfiniBand**

In addition to the LDT I/O Bus, there are several other new proposed bus standards attempting to met the needs of next generation, high performance systems. Chief among these are the Motorola proposal, RapidI/O and the communications oriented InfiniBand protocols. RapidI/O, like LDT, is oriented to "in-the-box" communications between processors, memory, I/O and communications channels. InfiniBand is broader in scope and applies a high level network architecture to the problem of inter-system

communications. While some InfiniBand proponents may offer intra-box solutions, the primary thrust of the InfiniBand effort seems more targeted to perform as a System Area Network or SAN, providing the means of clustering many separate systems together to form a highly available networked-based system. Rather than competing with InfiniBand, LDT will complement it, providing the high speed data paths in the box that can easily support the high bandwidth requirements of SANs.

### LDT I/O and InfiniBand Based System



**Figure 2 – InfiniBand protocols are designed for high speed interconnection between subsystems comprising a System Area Network. LDT I/O complements InfiniBand by providing a high speed link between subsystem and InfiniBand networks.**

RapidI/O, created by Motorola and proposed as an open standard to the industry, is similar to LDT in that it provides a high speed mechanism for inter-communications between on-board subsystems. By design, RapidI/O avoids high level protocols and supports only memory mapped I/O schemes. In addition it tries to anticipate system design requirements and limits the specification to provide only limited functionality in key areas. RapidI/O is defined as a three layer architectural hierarchy: a logical layer defines protocol and packet formats; a transport layer defines the routing information to move a packet from point to point; and a physical layer defines packet transport mechanisms, flow control, electrical characteristics and low level error management.

The RapidI/O architecture supports memory mapped distributed memory systems with a message passing protocol and an optional globally shared distributed memory programming model. Packets can contain a variety of data sizes. The complexity comes in when a local device attempts access of non-locally owned memory. The access must be controlled using a software maintained coherency method or a local device controlled message passing interface. This adds software complexity or additional hardware to the system. While the protocol and packet formats are independent of the actual system implementation, and may theoretically be used over serial or parallel interfaces, the standard is optimized for parallel byte-size data. It is anticipated that initial implementations will consist of 8- and 16-bit parallel point-to-point devices. The electrical interface will be the LVDS standard targeted towards short distance on-board links. With a frequency range from 250MHz to 1GHz, a maximum of 4 Gbytes/second can be achieved using a 16-bit interface.

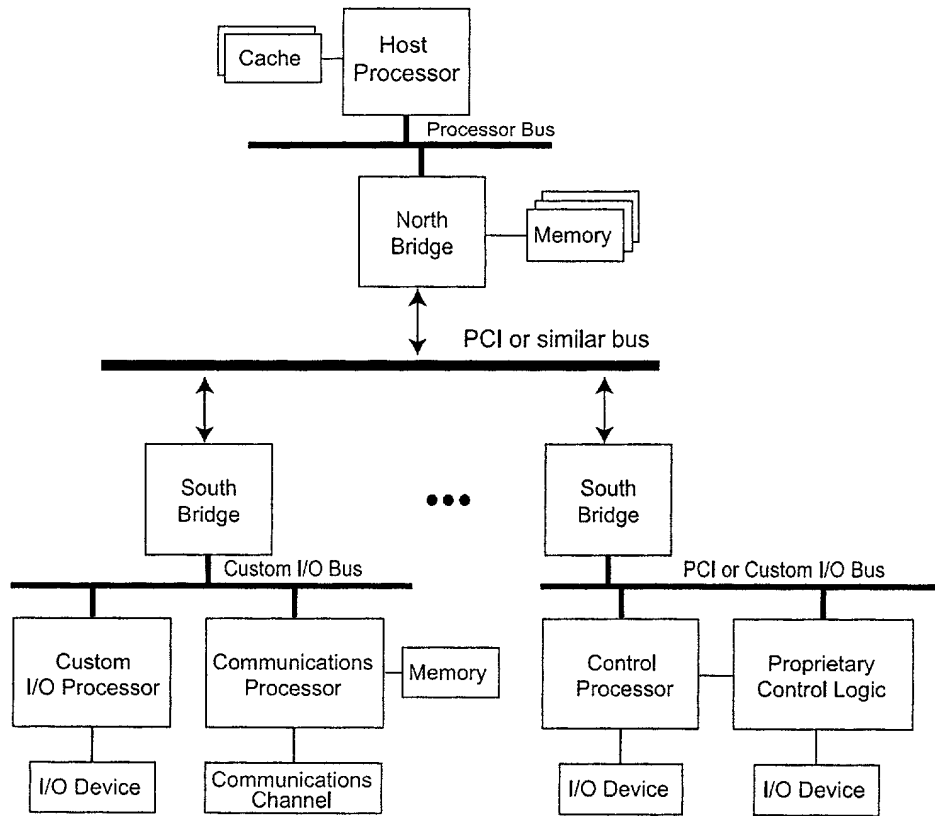
Like LDT, RapidI/O is intended to complement the InfiniBand SAN-oriented protocols. LDT provides a greater range of flexibility (using 2 to 32-bit wide configurations) and a greater bus bandwidth (up to 12.8 Gigabytes/second).

### **LDT System Design**

The foundation of the LDT I/O bus is dual point-to-point unidirectional links consisting of data path, control signals and clock signals. Each data path can be from two to 32 bits wide with standard bus widths of 2, 4, 8, 16 and 32 bits. Commands, addresses, and data share the data path. A link consists of the data path, a control signal and one or more clock signals. A complete LDT-based system consists of a processor with LDT port, the LDT bus, i.e., an input link and an output link, and any I/O channels connected to the LDT bus.

Traditional system bus architectures have a processor, a north bridge with memory and a mezzanine bus (usually a PCI bus) connected to additional PCI or custom I/O buses as shown below.

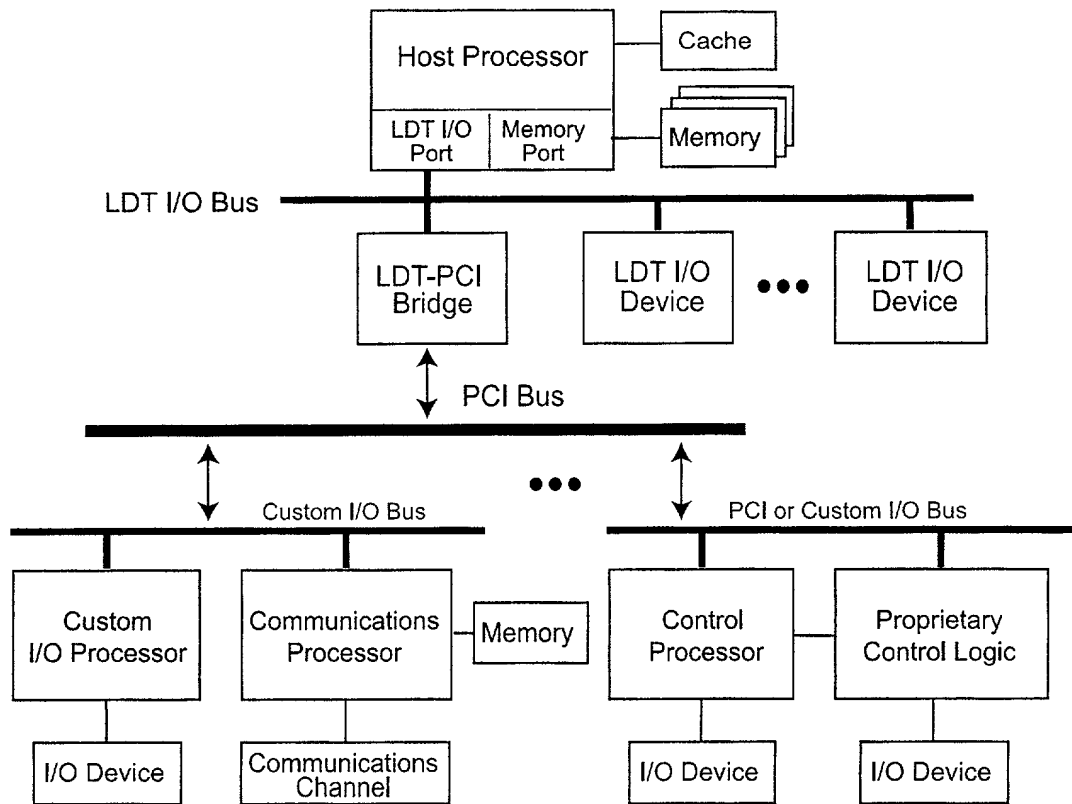
### Traditional Processor, Memory and I/O Bus Structures



**Figure 3 – A traditional north bridge/south bridge/PCI mezzanine bus structure.**

A much higher bandwidth LDT I/O-based system could be configured as shown below.

## LDT I/O-based Processor System



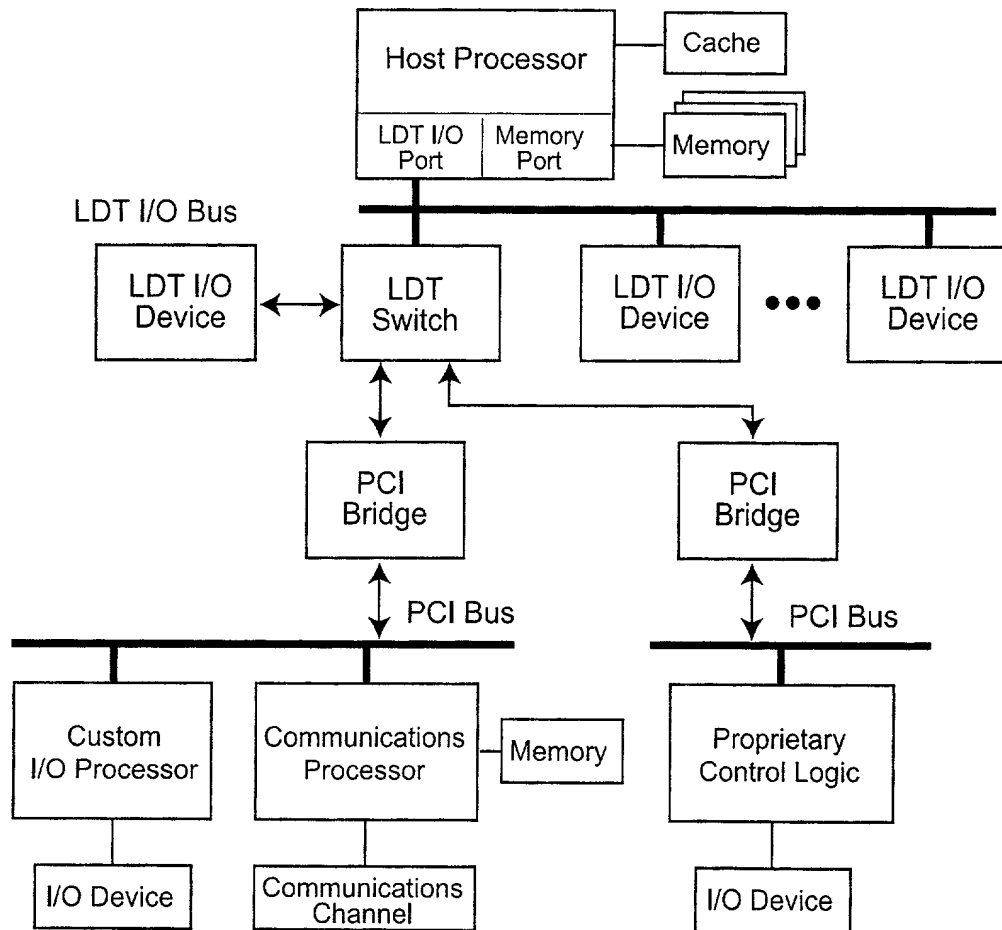
**Figure 4 – The LDT I/O bus can be used to connect directly to high speed LDT I/O devices and to bridge to slower peripheral buses such as PCI. The LDT bus uses the same tree of buses structure as PCI.**

The LDT I/O bus provides both point to point links and a scalable network topology using LDT I/O switching fabrics. As shown below, an LDT-based system can be expanded using LDT switches to support multilevel, highly complex systems.

An LDT I/O Switch handles multiple LDT I/O data streams and manages the interconnection between attached LDT I/O devices. A four port LDT switch could aggregate data from multiple downstream ports into a single high speed uplink, or it could route port-to-port connections.



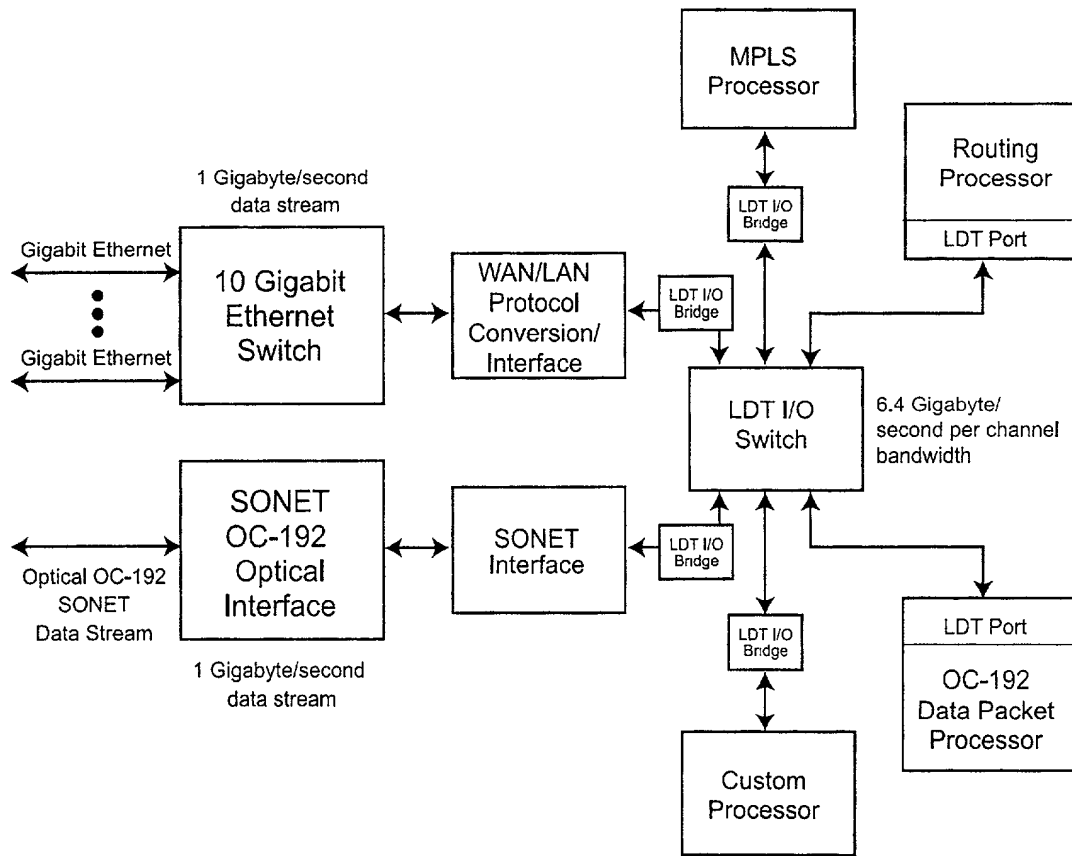
## Expanded LDT I/O-based Processor System



**Figure 5 – Using a LDT I/O switch, a LDT-based system can support multiple LDT I/O devices and link multiple high speed data paths while simultaneously supporting multiple slower speed buses.**

In this system, the LDT I/O bus enables the processor to talk directly to multiple LDT I/O devices on the bus and the LDT I/O switch enables connection to additional LDT I/O devices as well as to PCI-based subsystems through a PCI bridge.

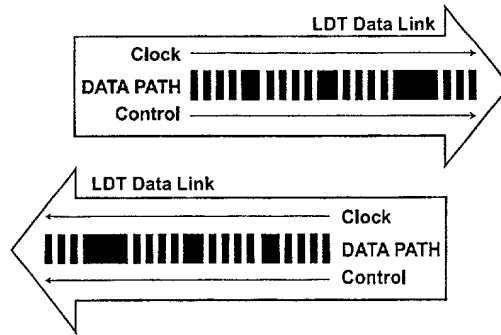
Using LDT I/O as the building block, even very high speed systems such as an OC-192 WAN to Gigabit Ethernet LAN integration system can be developed. Such a system could be configured as shown below.



**Figure 6 – The LDT I/O bus enables the design of high bandwidth optical switches and OC-192 WAN to LAN interfaces. The 10 Gigabit/second data stream of both OC-192 and 10Gb Ethernet can easily be routed through the system. A 16-bit LDT I/O bus can support 25.6 Gigabit/second bandwidth, enough for two OC-192 data streams.**

In this configuration, a very high-speed (1 Gigabyte/second and up) data stream from 10 Gigabit/second Ethernet LANs can be connected to the high speed Internet backbone and to the WAN OC-192 Optical SONET standard at 10 Gigabit/second data rates. To smoothly connect LANs to the WAN Internet backbone, protocol conversion from Ethernet packet data to SONET protocols must be performed. The high speed network processors in the system can perform this protocol conversion as well as performing other networking tasks such as data routing, address lookup, SONET data processing, and MPLS tagging. The LDT I/O bus provides the high-speed data throughput required to connect all of the diverse processors to the high speed data stream, enabling wire-speed performance through the LAN/WAN interface.

## LDT Basics



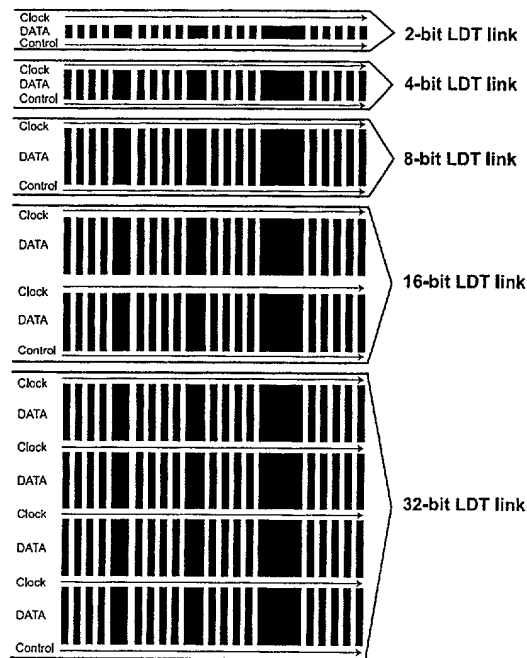
**Figure 7 – The LDT I/O bus consists of two unidirectional, point-to-point links, each consisting of 2, 4, 8, 16, or 32-bit data paths, a single control signal and one or more clock signals.**

An LDT I/O link is unidirectional, and in each direction can be 2 to 32 bits of data, with a single control signal and one or more clock signals. The data path supports standard bus widths of 2, 4, 8, 16, and 32 bits. Commands, addresses and data all share the data bus. The control signal indicates whether the information on the bus is a command or address or data. For each 8-bits or less of width, there is a forwarded clock signal. Information is transferred on both the rising edge and falling edge of the clock. Thus, a standard LDT I/O clock of 400 MHz yields an 800 MHz data rate per bit. By the end of 2001, clock frequencies for LDT devices are expected to at least double, providing a 1.6 Gbit/second data throughput per pair. This will enable a high-end LDT point to point bandwidth of 51.2 Gbits/second or 6.4 Gbytes/second.

LDT I/O Bus Characteristics					
Bus Type	Bus Width	Bandwidth	Clock Speeds	Data Protocol	Signal Type
Dual uni-directional links	2, 4, 8, 16, or 32 bits	3.2 to 6.4 Gigabytes /second.	400 MHz, moving to 800 MHz in 2001/2.	Packet based, with all packets multiples of four bytes (32-bits). LDT I/O commands, addresses and data are contained in packets.	Low-voltage differential signaling with a 60 Ohms differential impedance

A full 32-bit implementation of the LDT I/O bus (32-bits in both directions) yields a 6.4 Gigabyte/second data rate in each direction. This provides the bandwidth that modern embedded processors require for tasks such as large database manipulation (network routers), multimedia processing (digital consumer products such as set-top boxes and

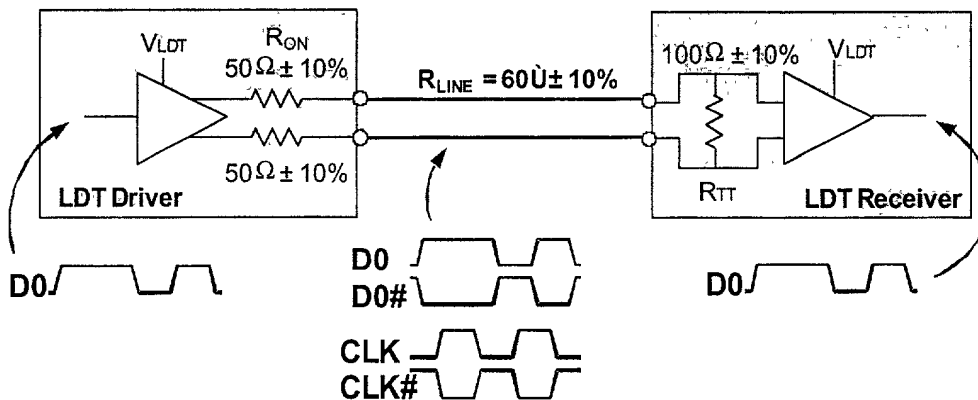
game consoles), and compression algorithms and communication protocol conversions for communication systems.



**Figure 8 – The LDT bus supports standard data widths of 2, 4, 8, 16 and 32 bits. For each 8-bits or less of bus width, a clock signal is used. For each link, a single control signal indicates whether the data payload is a command, address or data packet. Every 8-bit or less of bus width is paired with a forwarded clock signal.**

With a forwarded clock signal for each 8 bits or less of data width, the clock skew between the reference clock and bus signals is greatly reduced. Using multiple forwarded clocks for wider implementations greatly simplifies system design and debug of high speed functions. While a full 32-bit implementation yields an aggregate 6.4 Gigabyte/second bandwidth, a simpler, 2-bit implementation (2 bits in each direction) delivers a 400 Mbytes/second bandwidth in each direction, almost four times the peak bandwidth of PCI 32/33.

All LDT I/O commands, addresses and data are transported in packets. Packet lengths are multiples of four bytes and if the link is narrower than 32 bits, successive bit times are used to complete individual packet transfers.



**Figure 9 – The LDT bus uses low-voltage differential signaling at 1.2 volts, with a differential impedance of 60 Ohms.**

LDT I/O uses low-voltage differential signaling with a differential impedance of 60 ohms for command, address, data, clock, and control signals. The driver supply voltage is 1.2 volts. Differential signaling and the specified impedance provide a robust signaling system for use with low-cost printed circuit boards required for many embedded systems. Standard four-layer PCB material with no special layer techniques is sufficient for routing LDT interconnections. This is significant in the cost-sensitive manufacturing environment because special PCB material stackups and additional layers can easily triple the PCB material cost for a motherboard. The 50-ohm impedance and differential signaling also permit trace lengths up to 24 inches, and they span board interconnects well.

There is an ongoing effort with major connector companies to define a variety of connectors that can fit different price/performance parameters. More information on this topic will be available in 2001.

Differential signaling affects the number of pins in a LDT bus since it requires two pins per bit. However, pin count is reduced using LDT because of two other factors. First, because the LDT operates at higher clock frequencies, fewer pins can deliver better bandwidth than competing approaches. Second, differential signaling provides a return current path for each signal, which greatly reduces the number of  $V_{LDT}$  power and ground pins needed for a given package.

As shown in the table, in addition to command, address, data, clock and control pins, each LDT device will require  $V_{LDT}$  power, ground, PWROK (Power Okay) and RESET\_L (LDT Reset) pins.

LDT Pin Requirements					
Bus Width (Each Way)	2	4	8	16	32
Data Pins (total)	8	16	32	64	128
Clock Pins (total)	4	4	4	8	16
Control Pins (total)	4	4	4	4	4
High Speed Subtotal	16	24	40	76	148
VLDT	2	2	3	6	10
GND	4	6	10	19	37
PWROK	1	1	1	1	1
RESET_L	1	1	1	1	1
Total Pins	24	34	55	103	197

The Power Okay and Reset pins are single ended because of their low frequency use. For a low cost implementation, system developers can use the 2-bits in each direction LDT bus and achieve a 400 Mbytes/second LDT I/O bus with just 24 pins including V<sub>LDT</sub> and ground pins. This is by far the lowest cost implementation for I/O bus speeds in this range.

In some embedded applications, such as battery powered communications systems, digital entertainment devices, and handheld PCs, power consumption is a very critical design constraint. To support those applications, the LDT I/O bus has defined and LDTStop\_L (LDT Stop). This pin will put the LDT bus in a low-power state where virtually no power is used by the bus.

### Configuring LDT I/O Devices

LDT I/O devices with equal width transmitter and receiver links can be easily and directly connected. LDT I/O devices with transmit and receive links of different widths can also be easily and directly connected. Extra receiver pins are tied to logic zero while extra transmitter pins are left open. During power up, the RESET\_L is asserted and control is at logic zero, each LDT device transmits a bit pattern indicating the width of its receiver. Detection logic in each LDT device determines the maximum safe width for its transmitter and adjusts its output and receive channels to use only the safe maximum. This assures that LDT devices will communicate correctly before any software setup or configuration software is needed to optimally configure each device.

For systems using BIOS routines to configure I/O activity, LDT I/O devices use standard Plug 'n Play methodologies for exposing the control registers for BIOS routines to optimize the bus configuration. Other system firmware can be easily adapted. Enumeration of LDT devices and Bridges proceeds exactly like PCI enumeration using the same configuration header structures. AMD has registered the LDT Specific Capabilities Block with the PCI SIG. This encapsulates all the additional configuration information and settings needed for LDT. When a BIOS, or other firmware, enumerates the system's buses, this capabilities block will automatically be discovered as a part of

the linked list of other capability blocks for other devices. If a LDT I/O Host Bridge exists, the LDT I/O devices connected to the bridge can be enumerated as well. This protocol performs exactly like that used for PCI devices.

Drivers for LDT I/O devices will be unique to the devices, but they are no more complex than similar PCI type I/O device drivers. To make porting from PCI devices to LDT devices easier, the LDT I/O chain from a host bridge is enumerated like a PCI bus and devices and functions within an LDT I/O device are enumerated like PCI devices and functions.

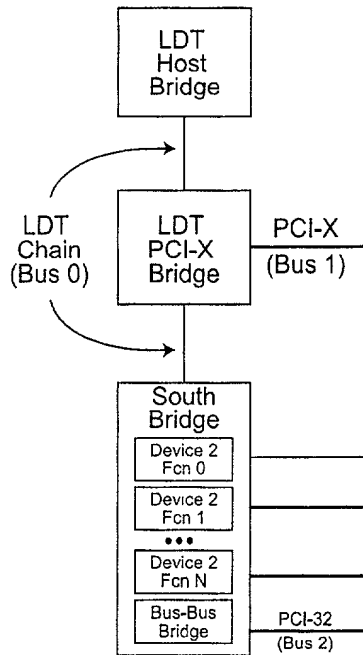
### **LDT I/O Bus Packet Protocols**

Communications between LDT I/O devices uses the metaphor of data streams. A LDT I/O bus can handle multiple data streams between devices simultaneously. LDT I/O devices can be daisy chained so that some streams may be passed through a node to the next node.

All data and commands are transmitted through a packet of information four bytes wide. Packets contain Source ID fields in the packet header and a data payload. There can be up to 32 IDs within a LDT I/O bus chain with some LDT I/O nodes containing multiple LDT I/O devices. Each LDT I/O node will determine if the information it is receiving is targeted to a device with the node. If not, the information packet is passed through to the next node. If an LDT I/O device at the end of the chain is not the target device, an error response is passed back to the Host Controller.

Commands and response from the Host controller have a Source ID of zero. Commands and responses sent from other LDT I/O devices have their own unique ID.

As shown in the figure below, a typical PC configuration could have a LDT Host Bridge daisy chained to a LDT to PCI-X Bridge and a South Bridge with multiple devices including a PCI-32 Bus Bridge.



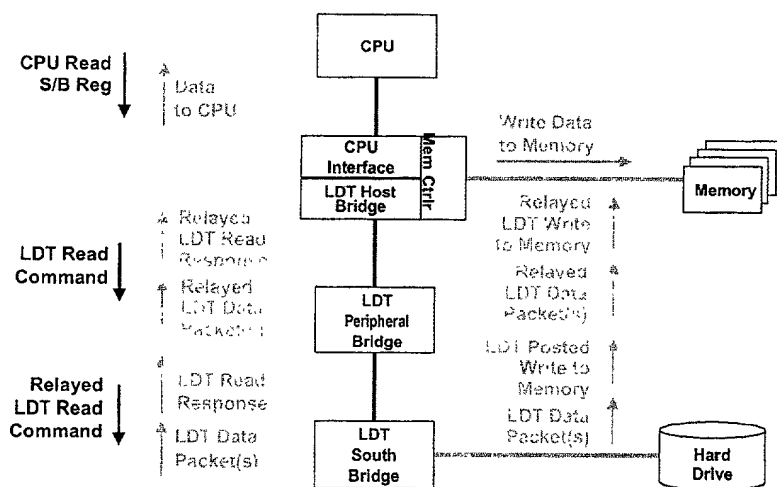
**Figure 10 – A typical LDT I/O bus chain can support multiple bus bridges as well as multifunctional LDT I/O devices.**

If a bus mastering LDT I/O device like a hard drive controller in the South Bridge sends a write command to memory above the Host Bridge, the command will be sent with the Source ID of the LDT device in the South Bridge. LDT permits posted writes such that the device does not wait for an acknowledgement of the write before proceeding. This is useful for large data transfers that will be buffered at the receiving end.



## CPU I/O Read

## Posted Master Write



Go to Messages Originate At Hard Drive Controller And Have Same Source ID

LDT transactions are divided into multiple virtual channels that have ordering semantics applied within those channels (but not between channels). LDT supports the PCI ordering and coherency model while allowing split transactions and more overlap between operations that are guaranteed to be unrelated.

LDT provides the same special handling as PCI for Read Responses with respect to Posted Writes. To illustrate, consider what would happen in the two examples above if the read request in the first example was to check a status register in the bus mastering LDT hard drive controller in the South Bridge that indicates whether the Posted Write to memory had completed. If the response to reading the status register can pass the Posted Write, the processor may be told the write had been completed before it actually happens and an immediate read at the memory location could yield the wrong data.

To prevent this, normally LDT does not allow Read Responses to pass Posted Writes within the same stream. A special bit in the Read Request, the pass posted write bit (PassPW), can be used to allow Read Responses to pass Posted Writes when it is safe to do so. In this way the handling of Read Responses can be accelerated.

Usually, LDT does not enforce ordering between streams. LDT I/O has special commands for forcing all streams to stay behind a fence command and to flush all commands from the chain. These are helpful in handling protocols for bridges to other common buses such as the AGP graphics bus.

### LDT I/O Command Syntax

All LDT I/O commands are four bytes long and begin with a six-bit command type field. The most common type of commands are Read Requests, Read Responses, and Writes. The remainder of the four bytes is command specific. The PassPW bit is located here.

Bit Time	7	6	5	4	3	2	1	0
0	Command Type							
1	PassPW	Command Specific Information						
2								
3								
4	-----							
5	Read and Write Command Address							
6								
7								

When the command requires an address, the last byte of the command is concatenated with an additional four bytes to create a 40-bit address.

A Write command or a Read Response command is followed by data packets. Data packets are four bytes to 64 bytes long in four-byte increments. The illustration below shows a packet of eight bytes.

Bit Time	7	6	5	4	3	2	1	0
0				Data [7:0]				
1				Data [15:8]				
2				Data [23:16]				
3				Data [31:24]				
4				Data [39:32]				
5				Data [47:40]				
6				Data [55:48]				
7				Data [63:56]				

Transfers of less than four bytes are padded to the four-byte minimum. Special sized reads and writes are supported with a four-byte mask field preceding the data. This is useful when transferring data to or from graphics frame buffers where the application should only affect certain bytes that may correspond to one primary color or some other characteristic of the displayed pixels. A control bit in the command indicates whether the writes are byte or doubleword size.

Reads and writes to PCI I/O space are mapped into a separate memory range eliminating the need for separate memory and I/O control lines or control bits in read and write commands. An additional memory range is used for in-band signaling of interrupts. A device signaling an interrupt performs a byte-sized write command targeted at the reserved memory space. The host bridge is responsible for delivery of the interrupt to the internal target.

### LDT I/O Implementations

LDT I/O supports a daisy chain connection scheme that enables LDT devices that are building blocks applicable to a wide range of embedded applications. For example a standard LDT I/O chipset could be used with a 32-bit embedded processor in a low-cost consumer application. The same chipset could be used in high-end applications in both the main system bus and auxiliary mezzanine I/O buses. Several embedded processor

technology companies are currently working on processors with LDT I/O ports for use in high-end communications and networking equipment.

### **Summary**

The LDT I/O bus architecture is an ideal solution to the next generation of embedded systems. It delivers up to 64 times the bandwidth of the 32-bit 33 MHz PCI bus commonly used to link I/O devices, yet is easy to implement. It also provides a scalable solution so that precise cost and space specifications can be matched to application requirements. It can be used in a wide variety of applications, ranging from desktop PCs and high-end servers to low-cost, low power embedded digital consumer applications and high performance network equipment – while maintaining compatibility with existing PCI-based software.

There are numerous companies working on the first generation LDT devices including API NetWorks, Inc. and AMD as well as several vendors using MIPS-based processor technology. The first silicon devices supporting the LDT I/O bus protocols should be appearing in the first quarter of 2001.

©2000 API NetWorks, Inc. All rights reserved.

For more information about API NetWorks, Inc., please visit the website at: [www.api-networks.com](http://www.api-networks.com).